

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-251383

(43)Date of publication of application : 06.09.2002

G06F 15/177

G06F 13/00

H04L 12/28

H04L 12/56

(71)Applicant : FUJITSU LTD

(72)Inventor : TAKAHASHI HIDEKAZU

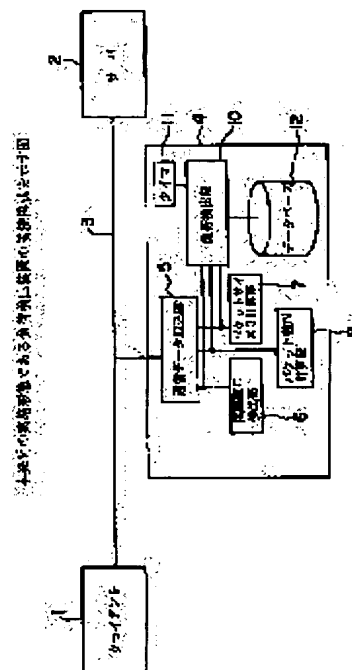
AOKI TAKESHI
YOKOYAMA KEN
KIKUCHI SHINJI

(54) NETWORK SERVER ALLOCATING DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To precisely and efficiently allocate a server.

SOLUTION: A device transferring data from a client to a plurality of servers through a network is provided with a repeating means changing an address of data transmitted from the client and transferring the data to one of servers, a connection managing means holding the corresponding relation of data and the servers and indicating the address to the repeating means and a server allocation means measuring and obtaining the processing capacity of the server, the client and a route, deciding the correspondence of data and the servers by using a function following a service distribution rate based on the capacity and transmitting the correspondence to the connection managing means.



LEGAL STATUS

[Date of request for examination] 06.12.2001

[Date of sending the examiner's decision of rejection] 05.08.2003

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's

decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2002-251383
(P2002-251383A)

(43) 公開日 平成14年9月6日 (2002.9.6)

(51) Int.Cl. ⁷	識別記号	F I	テマコード [*] (参考)
G 0 6 F 15/177	6 7 4	G 0 6 F 15/177	6 7 4 B 5 B 0 4 5
13/00	3 5 7	13/00	3 5 7 Z 5 B 0 8 9
H 0 4 L 12/28	2 0 0	H 0 4 L 12/28	2 0 0 D 5 K 0 3 0
12/56	1 0 0	12/56	1 0 0 Z 5 K 0 3 3

審査請求 有 請求項の数 4 O L (全 17 頁)

(21) 出願番号 特願2001-372976(P2001-372976)
(62) 分割の表示 特願平10-254318の分割
(22) 出願日 平成10年9月8日 (1998.9.8)

(71) 出願人 000005223
富士通株式会社
神奈川県川崎市中原区上小田中4丁目1番
1号
(72) 発明者 高橋 英一
神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内
(72) 発明者 青木 武司
神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内
(74) 代理人 100089244
弁理士 遠山 勉 (外1名)

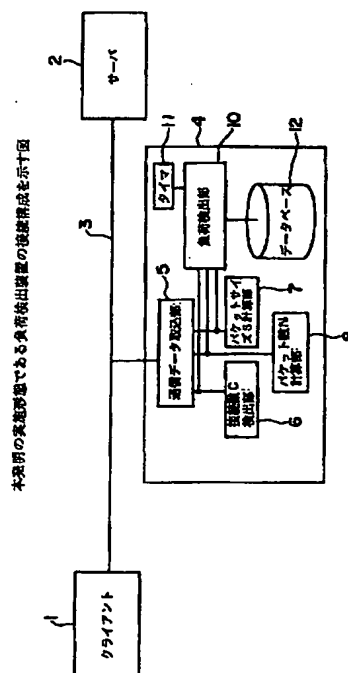
最終頁に続く

(54) 【発明の名称】 ネットワークサーバ割当装置

(57) 【要約】

【課題】 サーバの割当を高精度かつ効率的に行う。

【解決手段】 データをネットワークを介してクライアントから複数のサーバに転送する装置において、クライアントから送信されるデータを宛先を変えてサーバのいずれかに転送する中継手段と、データとサーバの対応関係を保持し中継手段へ宛先を指示する接続管理手段と、サーバ、クライアントおよび経路の処理能力を計測して求め、それに基づいたサービス分配率にしたがった関数を用いてデータとサーバの対応を決定し接続管理手段へ伝えるサーバ割当手段とを備える。



【特許請求の範囲】

【請求項1】 データをネットワークを介してクライアントから複数のサーバに転送する装置において、クライアントから送信されるデータを宛先を変えてサーバのいずれかに転送する中継手段と、データとサーバの対応関係を保持し中継手段へ宛先を指示する接続管理手段と、サーバ、クライアントおよび経路の処理能力を計測して求め、それに基づいたサービス分配率にしたがった関数を用いてデータとサーバの対応を決定し接続管理手段へ伝えるサーバ割当手段とからなるネットワークサーバ割当装置。

【請求項2】 前記請求項1のサーバ割当手段において、サーバの処理能力に応じた確率分布に対し、クライアントおよび経路の処理能力が低いほど一様分布に近づける修正を行うことで求めた修正確率分布を分配率とするネットワークサーバ割当装置。

【請求項3】 前記請求項1のサーバ割当手段において、現在サービス中のクライアントについてのクライアントおよび経路の処理能力の分布を求め、新規接続クライアントおよび経路の処理能力が分布に対して低いほど、サーバの処理能力に応じた確率分布を一様分布に近づけ、逆に高いほど各サーバの処理能力を際立たせる修正を行い修正確率分布を求め、それを分配率とするネットワークサーバ割当装置。

【請求項4】 前記請求項1において、サーバ割当手段を複数有しており、当該サーバ割当手段は、クライアントやサービス毎に選択されるネットワークサーバ割当装置。

【発明の詳細な説明】**【0001】**

【発明の属する技術分野】 本発明は、サーバ資源の割り当てに関するものである。

【0002】

【従来の技術】 近年、インターネット／イントラネットの急速な普及により、ネットワークサービスサーバの効率的利用およびサービス安定性が要求されてきている。サーバの効率的利用および安定したサービス供給にはサーバへのサービスの最適な割り振りが不可欠で、そのためにはサーバの負荷を正確に認識する必要がある。

【0003】 従来技術におけるサーバの負荷認識方法としては以下に示すものが知られている。

(1). エージェント方式

サーバ上にCPUやメモリなどの資源使用率を計測するプログラムをおく方式であるが、エージェント自身がサーバの負荷を上げ、エージェントが外部と通信を行う場合、そのために帯域を消費するなど、エージェントによる負荷計測精度への干渉が生じる。また、サーバへエージェントプログラムをインストールしなければならないため、汎用性に欠け指導コストが大きいという問題があ

った。

(2). 負荷計測通信方式

サーバに対しpingや擬似的なサービス通信などを行い、レスポンス時間などからサーバ負荷を求める方式であるが、計測のための通信で経路の帯域を消費してしまい、サーバも応答のための負荷を負うので負荷計測への干渉が生じる。また、計測に用いるプロトコルなどをサーバがサポートしている必要があり汎用性に欠けるといった問題があった。

(3). VC数、接続時間、接続頻度、接続エラー率、レスポンス時間の計測

これらは、クライアントからのパケットをサーバへ中継する装置上にあつて、中継時に計測したサーバへのVC数、接続時間、接続頻度、接続エラー率、レスポンス時間からサーバ負荷を求める方式であるが、接続時のサーバの振る舞いに基づくため誤差が大きい。精度を上げるためには多量の接続を必要とするため、少ない接続で大量の通信を行うサービスには適さない。また、中継が必須であるため、計測装置のスループットでサーバのスループットが制限されるといった問題があった。

(4). ヒット回数、ヒット率計算方式

WWWサーバなどへのパケットを調べ、アクセス対象であるファイルなどのコンテンツ毎にアクセス回数（ヒット回数）やアクセス頻度（ヒット率）を計測し、結果からサーバ負荷を求める方式であるが、アクセス対象ファイルを特定するためにはプロトコル毎のパケット解析処理が必要になり新規サービスに対応できない。さらにサーバの性能が既知でなければならない。サーバ性能をあらかじめ与えるにはサーバ性能をカタログ値や経験で求めるしかないが、サーバ性能はシステム構成や運用形態に大きく影響されるため、標準的な構成や形態に基づいたカタログ性能値は正確でなく、経験的に求める場合は少なくとも1回の障害を避けることができないといった問題があった。以上のように、いずれの方式もサーバに負担をかけずに高速かつ効率的にサーバの負荷を検出できるものではなかった。

【0004】 さらに、このようなサーバの負荷が正確に認識できないために、サーバで提供されるサービスを割り振ることも難しかった。サービスの割り振りという視点だけからは以下のような方式が提案されている。

(5). ラウンドロビンDNS方式

DNS(Domain Name System)サービスにおいて、1つのドメイン名に対し複数のサーバのIPアドレスを対応させるようエントリ表に設定しておき、クライアントからのサーバIPアドレスの問い合わせ要求に対し、各サーバをエントリ表にしたがい循環的（ラウンドロビン）に割り当て、割り当てられたサーバのIPアドレスを選択してクライアントに応えることで、サービスを複数のサーバへ分配する方式である。

【0005】 しかし、このラウンドロビンDNS方式で

は、サービスの分配率は、均等であるかあるいは単純な比率でしか行えず、各サーバは、それぞれの能力や動的な負荷状況に関係なく、割り当てられた分配率に応じてサービスを行わなければならないので、各サーバの負荷状況に差が生じてしまい、全体として非効率になってしまっていた。また、DNS問い合わせ情報は、通常クライアント側でキャッシングされてしまうので、比率変更が生じてそれをただちに反映できないという問題もあった。

(6). ハッシュテーブルを用いた分配方式

コネクションを管理するハッシュテーブルのエントリをサーバへ割り当て、割り当てるエントリ数に応じた比率でサービスをサーバへ分配する方式である。

【0006】この方式ではまず、クライアントからのサービス要求時にクライアントアドレスやサービスからエントリを決め、そのエントリが割り当てられているサーバへ要求を送る。そして、割り当てたエントリ数の比率に応じた数のサービスが各サーバへ振り分けられるので、高性能サーバへ多くのエントリを割り振ったり、高負荷になったサーバへの割り当てたエントリをそれ程負荷の高くないサーバへ割り当て直したりすることでサーバの効率的利用を実現している。

【0007】しかし、このハッシュテーブルを用いた分配方式では、ハッシュエントリ数の比率をサービス分配率へ正しく反映させるためには、偏りのないハッシュ値を生成するハッシュ関数が必要であるが、一般にハッシュキー（クライアントアドレスやポート番号など）のあらゆる分布に対し偏りのないハッシュ値を生成するハッシュ関数を見つけることは不可能である。また、分配率の精度はハッシュエントリ数に比例するため、精度を上げるためには多数のエントリ数が必要となり、記憶資源（バッファ）を多く消費してしまう。そして、結果的にコネクション管理に使用できる記憶資源（バッファ）が少なくなってしまう、大量なアクセスを扱うことができないといった問題があった。

(7). サーバの状態や性能にしたがった分配方式

サーバに対しpingなどでレスポンス時間を計測したり、クライアントからのパケットを中継して、中継時に接続時間、接続エラー率などを計測してサーバの負荷の高低を予測したり、サーバ間の性能比率を予測する等して、負荷や性能比に応じた量のサービスを分配する方式である。

【0008】しかし、この方式では、クライアントの処理能力、クライアントまでの経路の長さや帯域幅などとは無関係にどのクライアントへのサービスも平等にサーバへ分配するため、サーバの利用効率を最大にできなかった。

【0009】たとえば経路の帯域幅が狭かったり長かったりして経路がボトルネックとなっているクライアントや処理能力の低いクライアントにとってサーバの性能

（特にスピード）差や負荷はサービス品質には現れてこない。

【0010】逆に、近くや高速な回線で接続しているクライアントあるいは処理能力が高いクライアントにとってサーバの性能差や負荷はサービス品質に大きく影響することになる。そこで全てのクライアントへのサービスを平等に振り分けようとする、クライアントにとって必要以上のサーバ資源を振り分けたり、不足したサーバ資源を振り分けたりせざるを得なくなってしまうといった問題があった。

【発明が解決しようとする課題】以上述べたように、従来技術におけるサーバの負荷認識方法ならびにサーバ割当方法はいずれも問題があった。

【0011】本発明は、上記問題点を解決するべく提案されたものであり、その目的は、サーバの負荷をサーバに負担をかけずに高速かつ効率的に認識するとともに、サーバにおいて動的な負荷状況に応じたサービス分配を行い、設定や調整で得たサービス分配率を正確にサービス分配に反映させるとともに、クライアント毎に必要なサーバ資源を見積もりながらサービスを分配することでサーバの利用効率を最大にすることにある。

【0012】

【課題を解決するための手段】本発明は、前記課題を解決するため、以下の手段を採用した。

【0013】本発明の第1の手段は、データをネットワークを介してクライアントから複数のサーバに転送する装置において、クライアントから送信されるデータを宛先を変えてサーバのいずれかに転送する中継手段と、データとサーバの対応関係を保持し中継手段へ宛先を指示する接続管理手段と、サーバ、クライアントおよび経路の処理能力を計測して求め、それに基づいたサービス分配率にしたがった関数を用いてデータとサーバの対応を決定し接続管理手段へ伝えるサーバ割当手段とからなるネットワークサーバ割当装置である。

【0014】サーバの性能・負荷およびクライアント側の性能・負荷を計測して求めた方法にしたがってサービスを分配するので、サーバの動的な負荷状況の変化に自動的に対応でき、さらに、クライアントから見えるサービス品質を維持するために必要なだけのサーバを割り当てることができ、サーバの利用効率を最大にするという効果がある。さらに、サーバ割当決定を関数を用いて行うので、サービス分配率を正確に分配に反映できる効果がある。さらに、単一の接続管理手段のみで十分な効果を得られる。

【0015】本発明の第2の手段は、前記第1の手段でのサーバ割当手段において、サーバの処理能力に応じた確率分布に対し、クライアントおよび経路の処理能力が低いほど一様分布に近づける修正を行うことで求めた修正確率分布を分配率とするネットワークサーバ割当装置である。

【0016】クライアントおよび経路の処理能力の高さとサーバの処理能力がサービス品質に与える影響の大小の比例関係をサービス分配率へ反映するので、サーバ処理能力の影響がサービス品質に与える影響が大きいクライアントへ処理能力が高いサーバを優先的に割り当てることのできる効果がある。

【0017】本発明の第3の手段は、前記第1の手段でのサーバ割当手段において、現在サービス中のクライアントについてのクライアントおよび経路の処理能力の分布を求め、新規接続クライアントおよび経路の処理能力が分布に対して低いほど、サーバの処理能力に応じた確率分布を一様分布に近づけ、逆に高いほど各サーバの処理能力を際立たせる修正を行い修正確率分布を求め、それを分配率とするようにした。

【0018】サービス分配率を現在サービス中のクライアントおよび経路の処理能力分布との関係で調節するので、遠地からと近地からのクライアントの比率が変化するような場合でも自動的に対応できる効果がある。

【0019】本発明の第4の手段は、前記第1の手段において、サーバ割当手段を複数分であり、接続管理手段の規模はサービス分配に依存しないため、記憶資源（バッファ）の利用効率を上げる効果がある。

【0020】サービス別やクライアント別に割当対象サーバ群を使い分けたりサービス分配ポリシーを切り替えたりするなど多様なサーバ割り当てを単一の装置で行える効果がある。本発明の第5の手段は、クライアントからサーバへの通信を監視し、接続当たりの通信データサイズをサーバの負荷として計測するステップと、接続当たりの通信データサイズの変化を検出し、最大値を記録するステップと、前記最大値に対するその時点での接続当たりの通信データサイズが小さくなればサーバが高負荷であると判断するステップとからなるネットワークサーバ負荷検出方法である。

【0021】ここで、TCP等では、サーバはクライアントから送られたパケットデータを保持するための記憶資源（バッファ）を接続毎に均等に割り当てているが、サーバは次の受信で記憶資源（バッファ）に保持できるデータサイズをクライアントへ通知し、クライアントはサーバから通知されたサイズのデータをサーバに送るようになっている。

【0022】したがって、サーバが高負荷になるとクライアントから送られたデータをただちに処理できなくなるので、データの全てあるいは一部がサーバの記憶資源（バッファ）内に残留することになり、結果としてサーバは記憶資源（バッファ）内の残留データの分だけ小さいサイズをクライアントに通知せざるを得なくなる。

【0023】したがって、通信回線上での接続時間当たりでのデータサイズを検出することによって、サーバの高負荷状態を検出することが可能となる。本発明の第6の手段は、前記第5の手段において、監視通信最小数お

よび監視最小時間を用いて、監視した通信の数が監視通信最小数に達し、かつ、計測時間が監視最小時間に達するまで、接続数および通信データサイズを計測するようにした。

【0024】本発明の第7の手段は、前記第5の手段において、接続開始および接続終了の通信を認識し、接続開始および接続終了の通信データサイズを負荷検出対象から除外するようにした。

【0025】接続開始と終了の通信データは小さくサーバの負荷には依存しないので、通信総データサイズ計上から除外することで、負荷計測および高負荷判断の精度を上げる効果がある。

【0026】本発明の第8の手段は、前記第5の手段において、接続開始通信の情報を接続終了または接続確立まで保持するステップと、クライアントが接続失敗と判断して行う再接続のための接続開始通信を前記保持された情報に基づいて検出するステップと、接続開始通信回数に占める再接続通信の割合をサーバの負荷とし、この割合が高い場合にサーバが高負荷であると判断するものである。

【0027】サーバの負荷が大きい場合には、サーバはクライアントからの接続要求に回答通知を返信しなくなる。これに対してクライアントは接続要求を再送することになる。したがって、通信回線上でのクライアントの接続要求の再送を検出することによりサーバの高負荷を判定できる。

【0028】本発明の第9の手段は、前記第5の手段において、さらに、クライアントからの通信データサイズの分布を求めるステップと、前記分布からサーバの負荷に関係しない極端に小さな通信データを識別するステップと、前記極端に小さな通信データを負荷判定から除外するステップとを含むものである。

【0029】サーバ負荷に関係しない極端に小さな通信データを計測から除外することで負荷計測および高負荷検出の精度を上げる効果がある。本発明の第10の手段は、前記第5の手段において、クライアントからサーバへの通信から少なくともシーケンス番号を求めるステップと、前記シーケンス番号の最大値を、接続開始から終了まで保持するステップと、受信した通信のシーケンス番号を前記で保持されたシーケンス番号と比較するステップと、通信から得られたシーケンス番号が保持されたシーケンス番号よりも小さい場合、その通信を計測から除外するようにした。

【0030】シーケンス番号は通常昇順であるが、通信回線上での輻輳などによって通信の順序性の破壊もしくは欠損が起きると順序が昇順でなくなる。サーバは到着していないデータ以降のデータを処理できないので、サーバ負荷に関わらずサーバの受信可能データサイズが小さくなり、クライアントの通信データサイズも合わせて小さくなる。経路の影響を上記の方法で回避すること

で、サーバ負荷計測および高負荷検出の精度を上げる効果がある。

【0031】本発明の第11の手段は、前記第5の手段において、前記通信から得られたシーケンス番号が保持されたシーケンス番号よりも小さい場合、当該通信データを重み付け処理を行った後に計上する、または両シーケンス番号から経路上に問題がなかったときの通信データサイズを予測して、予測したサイズを負荷検出に計上するようにした。

【0032】本発明の第12の手段は、サーバからクライアントへの通信を監視し、サーバがクライアントへ通知する受信可能データサイズおよび接続数を計測するステップと、接続当たりの受信可能データサイズをサーバ負荷として求めるステップと、接続当たりの受信可能データサイズの最大値を記憶し、当該最大値に対する現接続当たりの受信可能データサイズが小さくなることでサーバが高負荷であると判断するネットワークサーバ負荷検出方法である。

【0033】本発明の第13の手段は、クライアントからサーバへの通信を監視し、サーバの負荷状態を検出するサーバ負荷検出装置であって、接続当たりの通信データのサイズを計算するデータサイズ計算手段と、接続当たりの通信データサイズの変化を検出し、最大値を記憶する記憶手段と、前記最大値に対するその時点での接続当たりの通信データサイズが一定値以下となったときにサーバの高負荷を検出する負荷検出手段とからなるネットワークサーバ負荷検出装置である。

【0034】

【発明の実施の形態】

【実施例1】図1は、本実施例1におけるサーバ負荷検出装置4の機能構成を示したものである。同図に示すように、サーバ負荷検出装置4は、クライアント1とサーバ2の通信回線3に接続されており、具体的には、ルータ等に実装することも可能である。

【0035】このサーバ負荷検出装置4は、同図に示すように、通信回線3を伝送されるパケットデータ（TCPパケット：Transmission Control Protocol Packet）を取り込む通信データ取込部5を有している。この通信データ取込部5には、接続数検出部6、パケット数計算部8およびパケットサイズ計算部7が接続されている。

【0036】接続数検出部6は、通信データ取込部5が取り込んだTCPパケットから単位時間当たりの接続数Cを検出する機能を有している。この接続数検出部6は、先頭パケットを意味するSYNパケットを検出すると+1とし、最後のパケットを意味するFINパケットを検出すると-1とする。これによって、当該サーバに現在接続されているクライアントの数が検出できることになる。

【0037】パケット数計算部8は、前記通信データ取

込部5が取り込んだ単位時間当たりのTCPパケットの数Nをカウントする機能を有しており、パケットサイズ計算部7は、前記通信データ取込部5が取り込んだ単位時間当たりのTCPパケットの合計サイズSを計算する機能を有している。

【0038】これらの各部の計算・計数データは、負荷検出部10に送られて後述の所定の演算処理に基づいて負荷が判定される。パケットサイズ計算部7によって計算されるパケット合計サイズSは、計測開始時に0とし、パケットが到着したらそのパケットサイズ分だけ順次増やしていく。なお、SYN、FINパケットについてはデータパケットに較べてそのサイズが小さく、サーバ負荷への影響が小さいため、無視してもよい。

【0039】パケット数計算部8によってカウントされるパケット数Nは、計測開始時に0とし、パケットが到着する度に+1とする。なお、ここでもSYN、FINパケットについては前述した理由によりカウントを無視してもよい。

【0040】パケット数計算部8によるカウントはNがある値Nminを超えるまで継続するが、このNminを超えても計測開始からの時間があらかじめ設定された時間Tminよりも短いときには、時間Tminが経過するまでカウントを継続する。

【0041】ここで、NminおよびTminはあらかじめパケット数計算部8に設定しておく。このようにNmin、Tminを併用することで、負荷検出のためのパケットのサンプル数が少ないために生じる計算誤差を減じることができ、また、サンプル数が多すぎるために生じるオーバーフローを回避することもでき、負荷検出精度を高めることができる。

【0042】負荷検出部10における負荷検出は以下の演算処理を行うことにより行われる。まず、負荷検出部10は、接続数検出部6から接続数Cを、パケットサイズ計算部7よりパケットサイズSを受け取ると、下記の式に基づいてサーバ負荷指標値Lを求める。

【0043】なお、ここでTはタイマ11により計測された計測時間である。ここで設定されたTmin経過時にサンプルとなるカウント数NがNminを超えているときには $T = T_{min}$ とする。

【0044】 $L = (S/C) / T$

ここで、Lは単位時間での1接続当たりのデータ転送量を意味することになる。このLを用いてサーバ2の負荷を検出することができる。

【0045】また、負荷検出部10では、サーバ2の処理能力限界予測値Lmaxを更新する。ここでLmaxは0を初期値とし、LがLmaxを超えた場合にはLmaxの値をLとする。ここでもし、LとLmaxとの間に以下の関係が成立すればサーバは高負荷であると判断することができる。

$$L < \alpha L_{max} \quad \text{ただし} \quad 0 < \alpha \leq 1 \quad \dots (1)$$

上式(1)において、 α はあらかじめ設定した定数である。図3は、前述した負荷検出部10における負荷検出をフロー図で示したものである。

【0046】まず、計測が開始されると、カウント数Nおよびサーバ負荷指標値Lはリセットされ、タイマ11がスタートされる(ステップ301)。そして、通信データ取込部5を介してパケットの受信が開始されると(302)、接続開始パケットSYNであるか(303)、接続終了パケットFINであるか(305)がそれぞれ判定される。ここで、接続開始パケットSYNである場合には、変数Vが+1される(304)。また、接続終了パケットFINである場合には変数Vが-1される(306)。

【0047】次に、新たなパケットが受信される度にNが+1されてサーバ負荷指標値Lが負荷検出部10で計算される(307)。この計算は前に説明した計算式に基づいて行われる。そして、前述の(1)式を用いて、サーバ負荷指標値Lが αL_{max} を超えている場合には、サーバは高負荷状態になっていると判定される。

【0048】このような高負荷判定は、タイマ値があらかじめ設定された T_{min} 以上となり、かつパケットのカウント数Nがあらかじめ設定された N_{min} 以上となったときに終了する(308)。

【0049】ここで、TCPでは、サーバ2はクライアント1から送られたパケットデータを保持するための記憶資源(バッファ)を接続毎に均等に割り当てている。サーバ2は次回受信で記憶資源(バッファ)に保持できるデータサイズをクライアント1へ通知し、クライアント1はサーバ2から通知されたサイズのデータを通信回線3を通じてサーバ2に送るようになっている。

【0050】したがって、サーバ2が高負荷になるとクライアント1から送られたデータをただちに処理できなくなるので、データの全てあるいは一部がサーバ2の記憶資源(バッファ)内に残留することになり、結果としてサーバ2は記憶資源(バッファ)内の残留データの分だけ小さいサイズをクライアントに通知せざるを得なくなる。

【0051】ここで、TCPはできるだけ大きなサイズのデータをやり取りするよう設計されたプロトコルであるため、サーバ2が高負荷となる前の状態では、クライアント1からサーバ2に送られるデータサイズは最大となり、その後、サーバ2の負荷が大きくなると通信回線3上を伝送されるデータサイズも小さくなる。本実施例では、図2に示すように、このデータサイズが小さくなることに着目してサーバの高負荷状態を検出している。

【0052】本実施例では、サーバ2が高負荷となる前の状態のときの通信回線3を伝送されるデータサイズが最大の値を L_{max} としてデータベース12に保持するようにしている。そして、(1)式で示したように、この L_{max} に定数 α を乗じた値(しきい値)とLとを比較

し、このLがしきい値以下となったときにサーバ2が高負荷状態であると判定している。

【0053】このように本実施例では、接続当たりのサイズを調べるため、接続数自体が減少することによるデータ合計サイズの減少で判断を誤ることを防ぐことができ、さらに α を用いることにより、外乱で生じるLの変動による高負荷誤検出を防ぐことができる。

【0054】なお、図4のフロー図は、図3で説明したフロー図とほぼ同様であるが、通信開始パケットSYNと通信終了パケットFINとを考慮しないで高負荷を判定する手順を示したものである。

【0055】

【実施例2】本実施例2は、クライアント1からサーバ2への再送処理を利用した高負荷検出方法である。

【0056】本実施例2で用いる装置構成は実施例1の図1で示したものとほぼ同様であるので図示は省略する。本実施例2では、個々の開始パケットSYNの情報をデータベース12に記録している(図6(a)~(c)参照)。そして、それぞれの開始パケットSYNの情報は、クライアントアドレス(IP)、クライアントポート番号(sp)、サーバポート番号(dp)の組で識別するようになっている。

【0057】TCPではサーバ2がクライアント1からの開始パケットSYNを受信すると、クライアント1に対してSYN受信確認パケットを返信する。ここで、クライアント1がサーバ2からのSYN受信確認パケットを一定時間経過しても受信できない場合、再度開始パケットSYNをサーバ2に対して再送信している。

【0058】この概念を示したものが図5である。同図(a)において、まずクライアント1aより接続要求(開始パケットSYN)がサーバ2に送信される。一方、別のクライアント1bからも接続要求(開始パケットSYN)がサーバ2に送信される。ここで、サーバ2のバッファ51に余裕のある場合、すなわち低負荷状態の場合には、サーバ2は、クライアント1aおよび1bに対して応答通知(受信確認パケット)を送信する。しかし、サーバ2のバッファ51に余裕のない場合には同図(b)に示すように、クライアント1からの接続要求(開始パケットSYN)に対して応答ができない。そこでクライアント1は、同図(c)に示すように、一定時間内にサーバ2からの応答通知(受信確認パケット)を受領できない場合には、サーバ2に対して接続要求を再送する本実施例2では、開始パケットSYNの数Csを接続数検出部6でカウントし、クライアント1からの開始パケットSYNの再送回数を検出してCsに対する開始パケットSYNの再送回数の比率Rsを算出し、これをサーバ負荷指標値Crsとする。

【0059】ここで、開始パケットSYNの再送は、開始パケットSYNから抽出したSYN情報がデータベース12に既に記録済みであれば再送であると判別でき

る。このことを示したのが図6である。同図(a)において、負荷検出装置4のデータベースには、SYN情報として、SYN1(IP1, sp1, dp1)、SYN2(IP2, sp2, dp2)およびSYN3(IP3, sp3, dp3)が記録されている。このときクライアント1からサーバに対して接続要求(開始パケットSYN4)が通信回線3を通じて発信される。負荷検出装置4は、この接続要求が、自身のデータベース12に格納されていない接続要求、すなわち初めての接続要求である場合には、この接続要求(SYN4: IP4, sp4, dp4)を当該データベース12に格納する(図6(b))

そして、この接続要求(SYN4)に対してサーバ2からクライアント1に返信通知がなされないときには、クライアント1よりサーバ2に対して当該接続要求(SYN4)が再送される。負荷検出装置4は、この接続要求(SYN4)を通信データ取込部5で取り込んで、負荷検出部10がデータベース12を検索することにより、既に自身が格納している接続要求であることを知り、その結果当該接続要求(SYN4)が再接続要求であると判定する。

【0060】負荷検出部10における具体的な計測方法は実施例1で説明した接続数CおよびパケットサイズSの計数・検出方法にしたがう。ここで、求めた開始パケットSYNの再送回数の比率Rs、すなわちCrsにより次式(2)が成立すればサーバ1は高負荷であると判定する。

$$Crs > \beta \quad \text{ただし、} 0 < \beta < 1 \quad \dots (2)$$

上式(3)において、 β はあらかじめ設定された定数である。

【0061】サーバ2は接続毎にクライアント1からのデータを保持するバッファ51を割り当てるが、割り当てバッファ51が枯渇すると接続を行わずに返信通知(SYN受信確認パケット)をクライアント1に返さない。そのため、クライアント1は開始パケットSYNの再送数の割合が増加することになる。したがって、式(2)よりサーバの高負荷を検出することが可能になる。図6(d)は、このような再送率(再送回数/通信回数)とサーバ負荷の関係を示したグラフ図である。

【0062】なお、上式(2)の定数 β は、外乱や瞬間的な高負荷状態による誤検出を防ぐためのものである。瞬間的な高負荷状態は生起確率が小さくかつ長くは続かないため、無視してよい。

【0063】

【実施例3】本実施例3は、負荷検出に際して、通信データサイズによって計測対象を弁別する技術である。なお、本実施例3も装置構成は図1と同様であるので、図1を用いて説明する。

【0064】本実施例3では、負荷検出部10において、クライアント1からのパケットサイズ S_i と D_s と

の間に以下の関係が成立する場合は S_i をパケット合計サイズLに加算しないで負荷検出を行う。

$$S_i < \gamma D_s \quad \dots (3)$$

ただし、 $0 < \gamma < 1$ 、 $D_s = f(S_1, S_2, \dots, S_{i-1})$ とする。

【0065】ここで、 γ はあらかじめ設定された定数である。 D_s は計測したパケットサイズの分布指標を求める関数で、たとえば平均値としてよい。また D_s の結果値が複数の値であるならば、重み付き加算や選択などによって単一の値としてもよい。

【0066】TCPではクライアント1は接続後、送信データサイズをサーバ2から通知されたデータサイズよりも小さなサイズにして送信を開始し、徐々に通知データサイズまで大きくしていく。そのため接続開始後間もないクライアント1からのパケットサイズはサーバ2の負荷に関わらず小さい。

【0067】したがって、接続開始後間もないクライアント1の数が多ければ多数の小さな送信データのために式(1)のLを小さく見積もってしまい負荷計測、高負荷検出の精度を落としてしまう可能性がある。

【0068】このことを概念的に示したものが図7である。同図(a)では、クライアント1aはサーバ2に対して比較的大きなサイズのパケットデータAを送信しているが、クライアント1bは、通信開始後間もないため、コマンドや応答信号など、比較的小さなサイズのパケットデータBを送信している。このような小さなパケットデータはサーバの負荷検出に際して無視して問題ない。

【0069】そこで、本実施例3では、式(3)を用いることにより、接続開始後間もないクライアント1からのパケットを検出してこれを計測対象から外すことで、負荷計測と高負荷検出の精度を上げている。

【0070】サーバが高負荷になれば、接続している全クライアントからのデータサイズが小さくなるが、データ保持のためのバッファ51の減少は比較的緩やかであるため、上のLの減少も緩やかである。また、全クライアントが一斉に新たに接続を開始することは確率的に低いいため、式(3)で十分である。

【0071】精度を上げるために式(3)に適用条件として D_s の下限値 D_{smin} を設定し、 D_s が D_{smin} 以下であれば式(3)を適用しない、つまり S_i をLに加算するようにしてもよい。

【0072】

【実施例4】本実施例4は、実施例1で説明した負荷検出において、通信回線上での輻輳等により生じるパケットの矛盾によりサーバ高負荷が誤検出されてしまうことを防止するための技術である。

【0073】本実施例4の装置構成は図1と同様である。ここで、クライアント1からサーバ2へのパケットは、クライアントアドレス(IP)とクライアントポー

ト番号 (sp) およびサーバポート番号 (dp) の組 (パケット識別子) およびシーケンス番号を接続開始から終了までがデータベース 12 に保持される。このとき保持されるシーケンス番号は最大値 (その時点での最終値) とする。

【0074】 負荷検出装置 4 がクライアント 1 からサーバ 2 へのパケットを受信したら、そのパケットからパケット識別子とシーケンス番号 P_i を求め、データベース 12 に保持している同一のパケット識別子のシーケンス番号 P_j と比較する。

【0075】 ここで、負荷検出部 10 の判定により、 $P_i < P_j$ が成り立てば、通信回線 3 上でパケットの追いつきが起きたか、途中のパケットが消失したことによって再送されたことがわかる。

【0076】 いずれにしても、このような状態でサーバ 2 が受け取るデータには途中で欠損が生じることになり、欠損個所以降のデータをサーバ 2 は処理することができず、欠損個所以降のデータはバッファ 51 に残留することになる。これによりサーバ 2 は受信可能なデータサイズが小さくなるが、原因はサーバ負荷ではなくクライアント-サーバ間の経路での輻輳などである。このことを概念的に示したものが図 8 である。同図では、クライアント 1 からサーバ 2 に対してパケットデータ「1~3」が送信されているが、これが経路輻輳等の要因でパケットデータ「2」のみが消失している。クライアント 2 は、受信したパケットデータ「1, 3」をバッファ 51 に格納する。ここで、クライアント 1 に対して応答通知 (パケットデータ「2」の再送要求) を送るが、自身のバッファ内ではパケットデータ「2」が受信されていないため、既に到着しているパケットデータ「3」以降を処理できない状態となっている。

【0077】 クライアント 1 はパケットデータ「2」に関する応答通知を重複して受信するとパケットデータ「2」の再送を行う。このようにして、パケットデータ「2~5」が揃うことによりサーバ 2 は受信したこれらのパケットデータを処理できる状態となるが、ただちに処理には移行できないため、クライアント 1 に通知するバッファの空きサイズは本来のバッファの大きさ N よりもはるかに小さい n となる。

【0078】 次に、クライアント 1 はサーバ 2 から通知されたサイズ n に格納可能なサイズのパケットデータ「6」を送信するが、実際にはこのパケットデータ

$$L3 = \alpha 3 \cdot L3_{\max} \quad \text{ただし、} 0 < \alpha 3 \leq 1 \quad \dots (5)$$

上式 (5) において、 $\alpha 3$ はあらかじめ設定された定数である。

【0087】 サーバ 2 は、クライアント 1 に対して、自身が処理できるバッファ 51 の空きサイズ、すなわちウィンドウサイズを通知しているが (図 9 (a))、ここで、サーバ 2 の負荷が上昇しクライアント 1 から送られたデータを完全に処理できなくなると、図 9 (b) に示

「6」を受信する段階では、パケットデータ「1~5」が処理されているため、バッファには広い空き空間が存在しており高負荷状態とはなっていない。

【0079】 つまり、本実施例 4 では、このような図 8 に示した状態を高負荷と判定しないようにしている。以上のような理由により、 $P_i < P_j$ が成立したパケット P_i は計測から除外する。あるいはある重み付けを行い計上してもよく、さらに負荷検出部 10 において、 $P_j - P_i$ をパケットサイズに加えて計上してもよい。

【0080】 ここで、 $P_j - P_i$ の算出は、サーバ 2 内のバッファ 51 に残留しているデータの予測サイズをパケットサイズに加えることでデータの欠損が生じなかった場合のサーバ 2 からクライアント 1 へ通知される受信可能データサイズすなわち現パケットのサイズの予測を行うことを意味している。

【0081】

【実施例 5】 本実施例 5 は、サーバ 2 がクライアント 1 に送信するパケットデータを監視することでサーバ 2 の負荷を判定するものである。

【0082】 本実施例 5 の負荷検出装置 4 は、サーバ 2 がクライアント 1 へ送るパケット中のウィンドウサイズ合計値 Sw と接続数 C とを監視する。ウィンドウサイズはサーバ 2 がクライアント 1 へ通知する受信可能なデータサイズである。

【0083】 C の値は、サーバ 2 からクライアント 1 への開始パケット SYN を検出したときに 1 増やし、終了パケット FIN を検出した場合 1 減じることで求める。ここで、 Sw と C の計測は実施例 1 と同様である。

【0084】 サーバ 2 の負荷指標値 $L3$ は次式で求められる。 T は実施例 1 の T と同様であるが必ずしも必須ではない。

$$L3 = (Sw / C) / T \quad \dots (4)$$

$L3$ は接続当たりのウィンドウサイズを意味する。 $L3$ を用いてサーバ 2 の高負荷を検出する方法は次の通りである。

【0085】 まず、サーバ 2 の処理能力限界予測値 $L3_{\max}$ を更新する。 $L3_{\max}$ は、0 を初期値とし $L3$ が $L3_{\max}$ を超えた場合に $L3_{\max}$ の値を $L3$ とすることで行う。

【0086】 ここでもし、 $L3$ と $L3_{\max}$ との間に以下の関係が成立すれば、サーバ 2 は高負荷であると判定する。

すように、サーバ 2 はクライアント 1 に対して以前より小さいウィンドウサイズ n を通知する (より具体的には次回受信可能なデータサイズである)。このように、サーバ 2 からクライアントに通知されるウィンドウサイズと時間の関係をグラフで示したものが図 9 (b) である。

【0088】 サーバ 2 の負荷は接続した全てのクライ

ントに影響するため、L3はサーバ負荷の上昇に伴い減少する。したがって、式(4)よりサーバ負荷を計測することができ、式(5)で高負荷を検出することができる。

【0089】

【実施例6】本実施例6は、本発明のサーバ割当装置をクライアント・サーバ間でTCPパケットを中継する装置として実現した場合である。

【0090】図10において、宛先変換・パケット中継手段1002は、クライアント1から受信したパケット1010が接続要求を意味する開始パケットSYNであると、サービスを割り当てるサーバを決定するためにサーバ割当手段1001のサーバ選択手段1007に対してサーバ割当指示1020を出力し、クライアント側処理能力計測手段1008に計測指示1021を行う。

【0091】サーバ処理能力計測手段1004は、各サーバの処理能力を計算し、結果データ1013をサーバ割当確率計算手段1006に送出する。各サーバの処理能力は、サーバ2に対してpingなどを発信してそれに対する応答時間から算出することもできるし、ユーザがあらかじめ設定しておいてもよい。また、本実施例1～5で述べたサーバ負荷検出装置を用いることもできる。

$$P_i = p_i / (p_1 + p_2 + \dots + p_n) \quad \dots (6)$$

確率修正関数Mは、図12に示すように、クライアント側の処理能力が低いほどP_{sD}を一様分布に近づけるように修正する関数となる。たとえばpingなどによる

$$P_i' = P_i + (P_{av} - P_i) \cdot \frac{2}{\pi \cdot \arctan(\alpha \cdot T_{ping})} \quad \dots (7)$$

ここで、P_{av}は、P_iの平均値であり、αはあらかじめ設定された0より大きい数である。また、arctan(x)はtan⁻¹(x)を意味する。

【0095】このP_i'から修正確率分布MP_{sD}を求める。サーバ割当確率計算手段1006は、求めたMP_{sD}をサーバ選択手段(1007)に送る。サーバ選択手段(1007)は、MP_{sD}から図13に示すテーブルを生成し、0～1の任意の値をとる一様乱数値を用いて実現する。同図のテーブルは、たとえばサーバ数の要素を持つ配列で実現し、各要素に0～1までの範囲P_iの最大値および最小値とサーバアドレスの組をおき、一様乱数値を含む範囲を持つ要素のサーバアドレスをサービス割当サーバアドレスとしてよい。ただし、各要素の範囲は他の要素の範囲と重複しないようにする。

【0096】P_{sD}およびMP_{sD}の確率分布については、サーバ処理能力値P_iおよびP_i'を度数分布として実現してもよい。この場合、一様乱数値は0から全P_iの合計値までの範囲をとるようにする。

【0097】サーバ選択手段1007は、割当サーバを決定したらそのサーバアドレス1012を接続管理手段

【0092】クライアント側処理能力計測手段1008は、中継手段1002からの指示があると、クライアント1および通信回線3の処理能力1018を計算し、サーバ割当確率修正情報生成手段1009に報告する。ここで、クライアント側処理能力は、たとえばクライアントに対してpingなどを発信してその応答時間から求めることもできる。また、B_{prob}などの帯域計測手法を用いたり、クライアント1の通信についての過去の記録、パケットから抽出されるウィンドウサイズやTTLから求めることもできる。

【0093】サーバ割当修正情報生成手段1009は、クライアント側処理能力1018から、サーバ割当確率分布に対する修正関数1022を生成する。図11にサーバ割当確率分布P_{sD}の例を、図12下図に修正関数M(1022)の例を示す。

【0094】サーバ割当確率計算手段1006は、サーバ割当確率分布P_{sD}に修正関数M(1022)を適用してサーバ割当確率分布MP_{sD}を求める。確率分布P_{sD}は、現時点で処理能力が高いサーバほど割当確率が高くなるような分布とする。たとえば、現時点での各サーバの処理能力値(後述)をp₁, p₂, ..., p_n(nはサーバの数)とすれば、サーバS_iへの割当確率P_iを次式で求めることもできる。

応答時間T_{ping}をクライアント処理能力とすれば、各サーバ処理能力P_iについて次の式から修正P_i'を求めてもよい。

1003に送る。接続管理手段1003は、宛先変換・パケット中継部より受け取った開始パケットSYNまたはその一部からクライアントアドレス(IP)、クライアントポート番号(sp)、宛先ポート番号(dp)の組情報を抽出し、組情報とサーバ割当手段1001とから受け取ったサーバアドレスの対を記録する。ここで、記録には組情報をキーとするハッシュテーブルを用いてもよい。接続管理手段1003は、宛先変換・パケット中継手段1002へサーバアドレス1012を送る。

【0098】宛先変換・パケット中継手段1002は、受信したクライアント1からのパケットの宛先を接続管理手段1003から受け取ったサーバアドレス1012に変換してサーバ2に送信する。

【0099】サービス中、宛先変換・パケット中継手段1002は、接続管理手段1003にパケット1014を送り、この接続管理手段1003は、パケット1014から求めた組情報より割当サーバアドレス1012を求めて宛先変換・パケット中継手段1002に送る。開始パケットSYNと同様に、宛先変換・パケット中継手段1002は、受信したクライアント1からのパケット

の宛先を、接続管理手段1003から受けたサーバアドレス1012に変換してサーバ2へ送信する。

【0100】サービス終了時、すなわち終了パケットFIN受信時はサービス中と同様であるが、これを受信した接続管理手段1003は、パケットに対応した組情報を破棄する。

【0101】本実施例では、確率分布を用いてサービス割当を決定することで、処理能力が高いクライアントほど処理能力が高いサーバを割り当て易くなるので、応答時間などのサービス品質に対するサーバ処理能力の影響力の大小にしたがったサービス割り当てが可能になる。

【0102】サーバ割当手段1001のサーバ割当確率修正情報生成手段1009は、過去のクライアント側処理能力の分布(図14)を求め、新規接続クライアント

$$p_i' = p_i + (P_{av} - p_i) * \beta * 2 / \pi * \arctan(\alpha * \delta_c + \gamma)$$

ここで、 P_{av} は p_i の平均値であり、 α 、 γ はあらかじめ設定された0より大きい数である。また、 $\arctan(x)$ は $\tan^{-1}(x)$ を意味する。 β は-1のとき、 $\delta_c < 0$ であり、 $-dp_j/p_j$ のとき $\delta > 0$ である。 p_j は p_i の最小値であり、 $dp_j = P_{av} - p_j$ となる。

【0103】このように、新規接続クライアントのクライアント側処理能力値の過去のクライアント側処理能力値の分布に対する隔たりにしたがってサーバ割当を行うことにより、各時点におけるクライアントに応じたサーバ割当が可能となる。たとえば、遠隔地からと近隣地からのクライアントの比率が時間帯によって変動する場合などに自動的に対応することが可能となる。

【0104】さらに、本実施例では、サーバ割当手段(1001)を複数配置して、それぞれをクライアントアドレス、クライアントポート番号、サービスポート番号などに応じて選択してもよい。

【0105】サービス毎やクライアント毎に割当対象サーバ群を使い分けたり、サービス分配ポリシーを切り替えたりすることが可能となり、多様なサービス割当を一つの装置で行うことができる。

【0106】

【発明の効果】以上説明してきたように、本発明は、サーバの負荷計測および高負荷検出をクライアント・サーバ間の通信を監視して行うので、サーバへ手を加える必要がなく、サービス以外のパケットを出さない。したがって、いかなるサーバへも対処でき導入コストが低く負荷への干渉が一切ないという効果がある。また、プロトコルに依存しない指標で負荷計測、高負荷検出を行うので、いかなるサービスへも対処できるという効果もあり、サービス中の通信状態を監視するために外乱の影響が小さく精度が高いという効果もある。

【0107】さらに、サーバの提供するサービスを複数

のクライアント処理能力値の分布からの隔たり δ を求め、新規接続クライアントのクライアント側処理能力値の分布からの隔たり δ_c を求める。そして δ_c を修正関数 M に加味して確率分布を修正する(図15)。ここで、たとえば δ_c を以下の式で求める。

$$\delta_c = P_{ca} - p_{ci}$$

ここで、 P_{ca} は過去のクライアント側処理能力平均値であり、 p_{ci} は新規接続クライアントのクライアント側処理能力値である。 δ_c が小さいほどサーバ処理能力値 p_i を全 p_i の平均値に近づけ大きいほど平均値からの隔たりを大きくするように修正関数 M を定める。ただし、平均値からの隔たりを大きくする場合、 p_i の修正値 p_i' が負数にならないようにする。たとえば、式(7)を以下のようにしてもよい。

$$\dots (7')$$

のサーバに分担させる際、サーバ構成の変更やサーバの状態の変化に対し、クライアントから見えるサービス品質に対するサーバ処理能力の影響の大小に応じて各サーバの負荷分担を自動的に、かつ、効率的に割り振るので、クライアントにとって、迅速なサービスの供給を受けることができる効果がある。

【図面の簡単な説明】

【図1】 本発明の実施形態である負荷検出装置の接続構成を示す図

【図2】 実施形態におけるサーバの高負荷判定を行うためのデータサイズと時間との関係を示したグラフ図

【図3】 実施例1におけるパケット監視方法を示すフロー図(1)

【図4】 実施例1におけるパケット監視方法を示すフロー図(2)

【図5】 クライアントからサーバへの接続要求と、バッファの状態に応じた応答処理を説明するための図

【図6】 クライアントからサーバへの接続要求の再送を説明するための図

【図7】 負荷検出においてデータサイズによって対象データとするか否かの弁別を行う例を示す説明図

【図8】 シーケンス番号に基づくサーバの処理を説明するための図

【図9】 サーバからクライアントへの通信を監視する説明図

【図10】 実施形態のサーバ割当装置の構成を示すブロック図

【図11】 サーバ割当確率分布 $P_s D$ を説明するためのグラフ図

【図12】 修正関数を説明するための図(1)

【図13】 実施形態において、サーバ選択手段により生成されるテーブルの一例を示す図

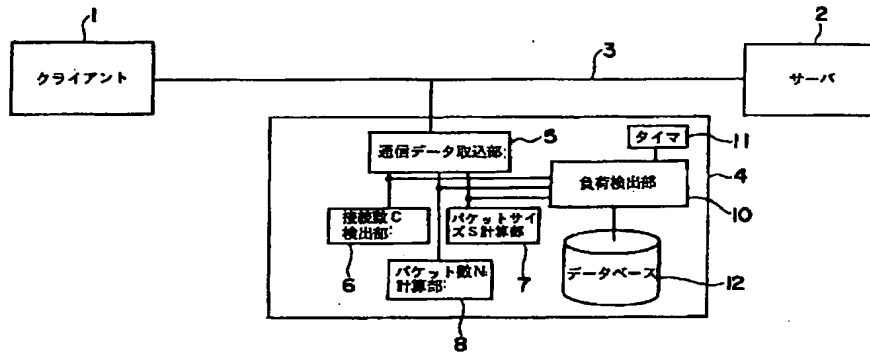
【図14】 過去のクライアント側処理能力値の分布例

を示すグラフ図

【図15】 修正関数を説明するための図(2)

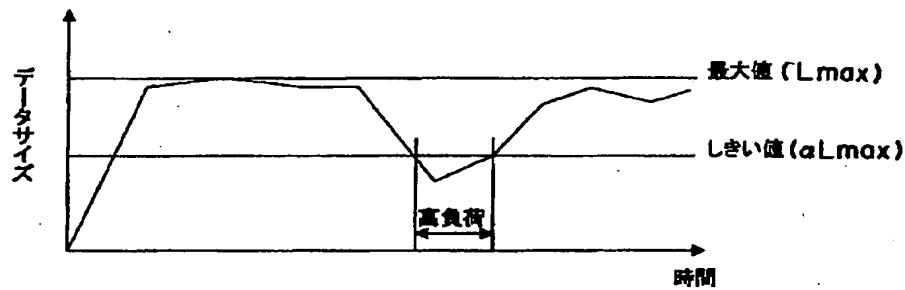
【図1】

本発明の実施形態である負荷検出装置の接続構成を示す図



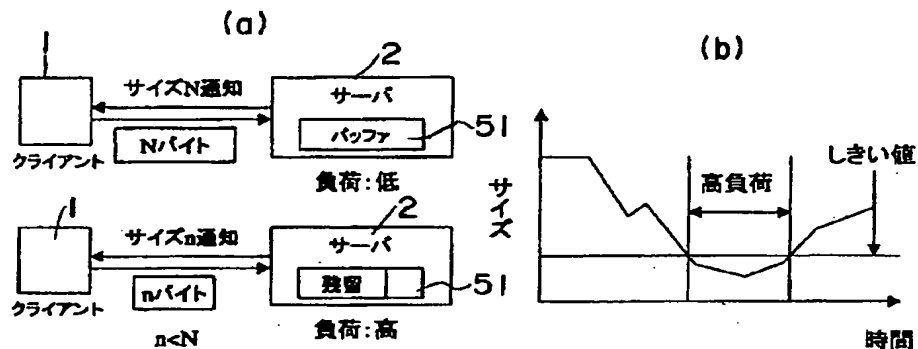
【図2】

実施形態におけるサーバの高負荷判定を行うためのデータサイズと時間との関係を示したグラフ図



【図9】

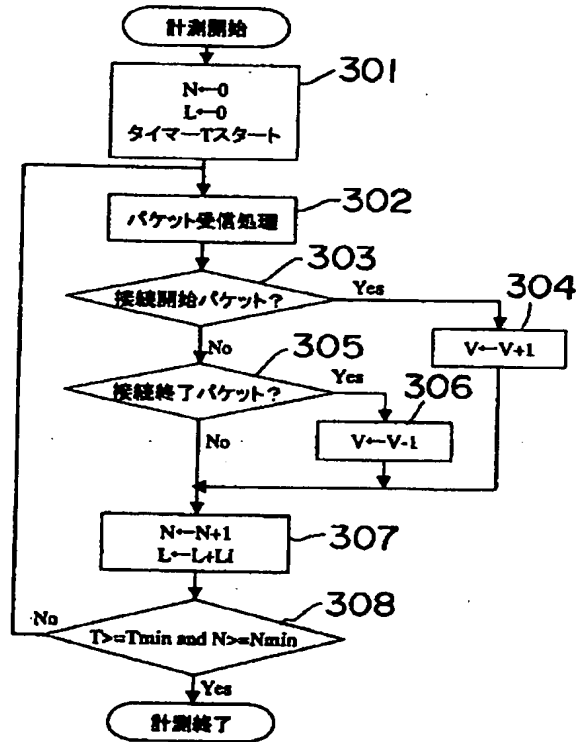
サーバからクライアントへの通信を監視する説明図



【図3】

【図13】

実施例1におけるパケット監視方法を示すフロー図 (1) 実施形態において、サーバ選択手段により生成されるテーブルの一例を示す図

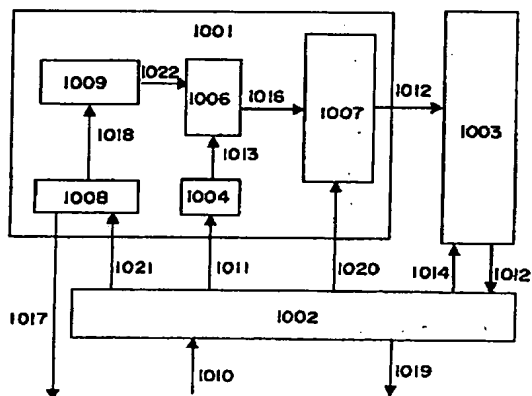


#0	#1	#2	#3	#4
0~0.04 サーバ1	0.05~0.34 サーバ2	0.35~0.55 サーバ3	0.56~0.80 サーバ4	0.81~1.00 サーバ5

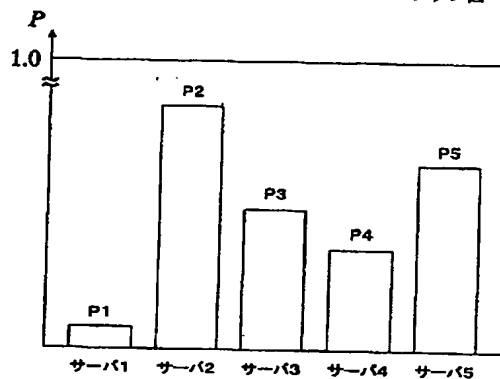
【図10】

【図11】

実施形態のサーバ割当装置の構成を示すブロック図

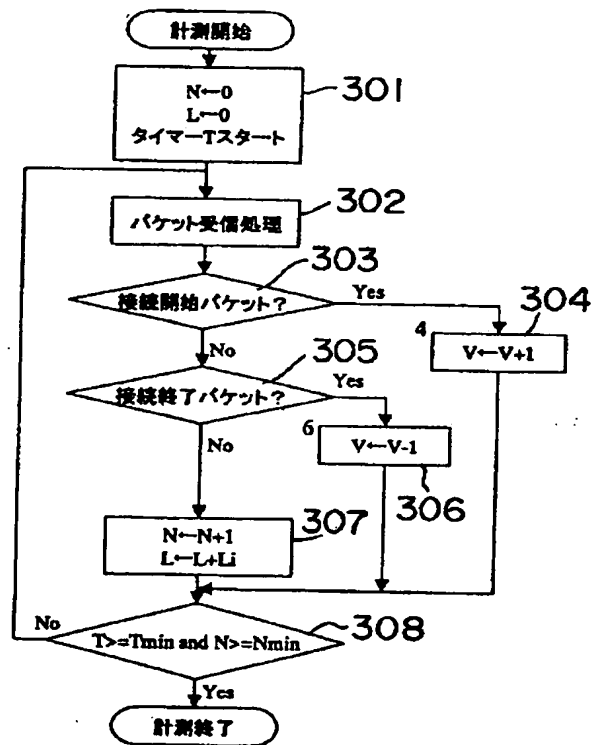


サーバ割当確率分布 P s D を説明するためのグラフ図



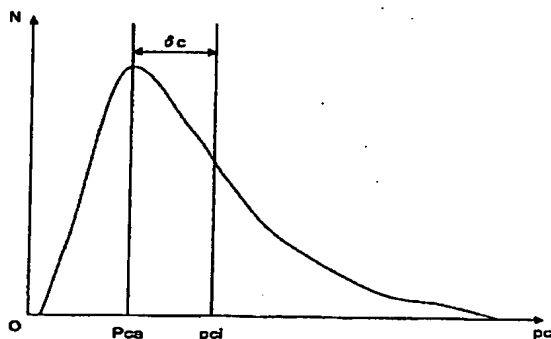
【図4】

実施例1におけるパケット監視方法を示すフロー図(2)



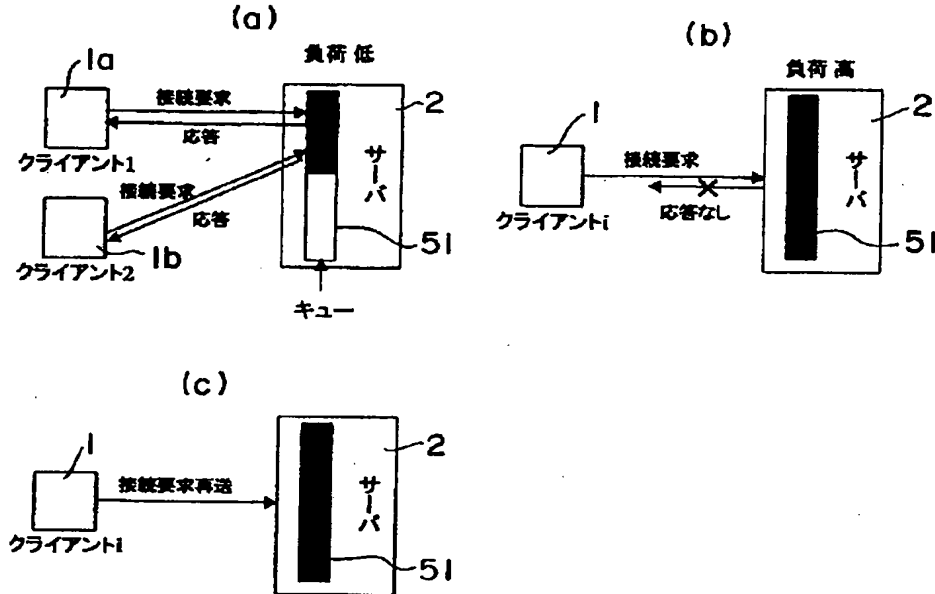
【図14】

過去のクライアント側処理能力値の分布例を示すグラフ図



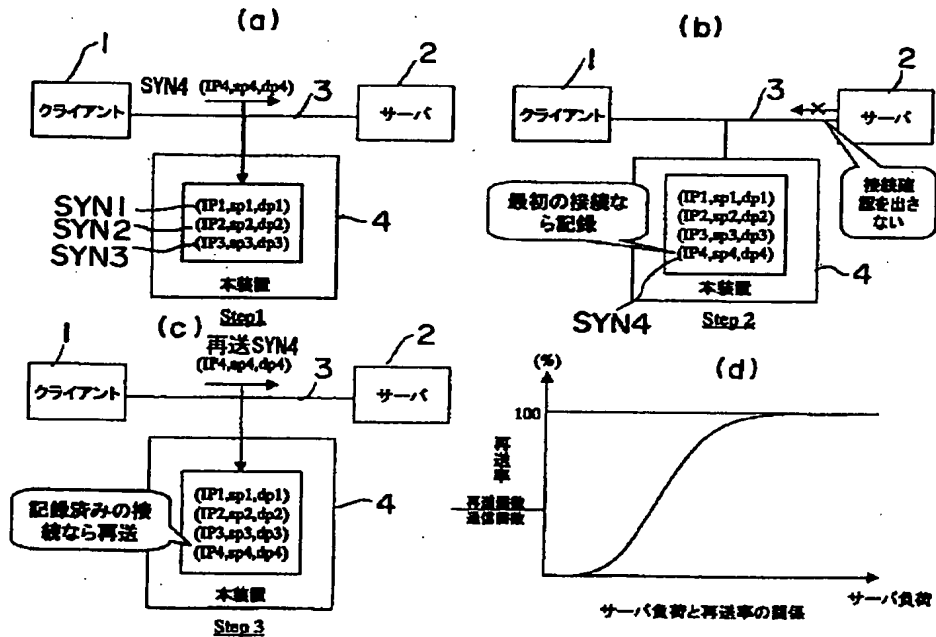
【図5】

クライアントからサーバへの接続要求と、バッファの状態に応じた応答処理を説明するための図



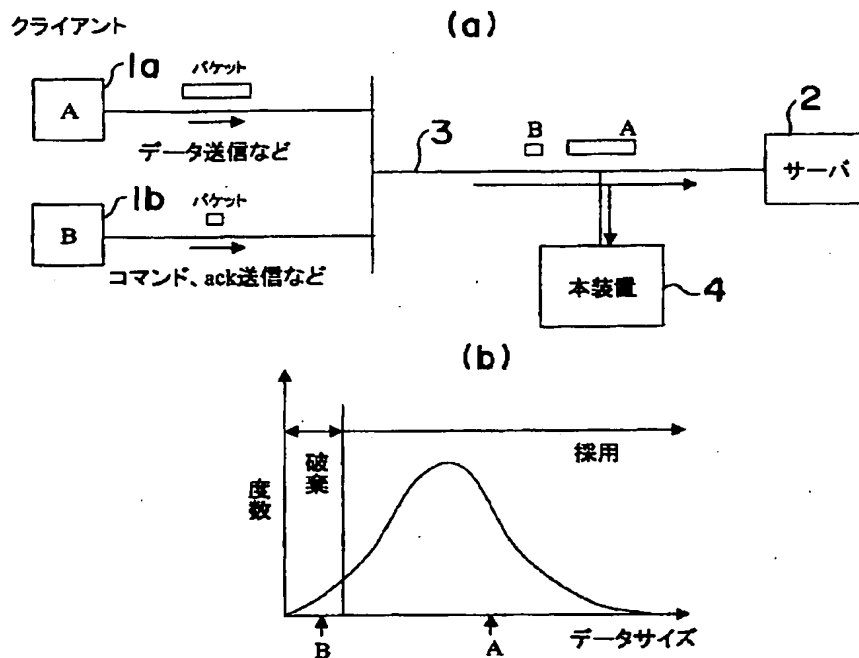
【図6】

クライアントからサーバへの接続要求の再送を説明するための図



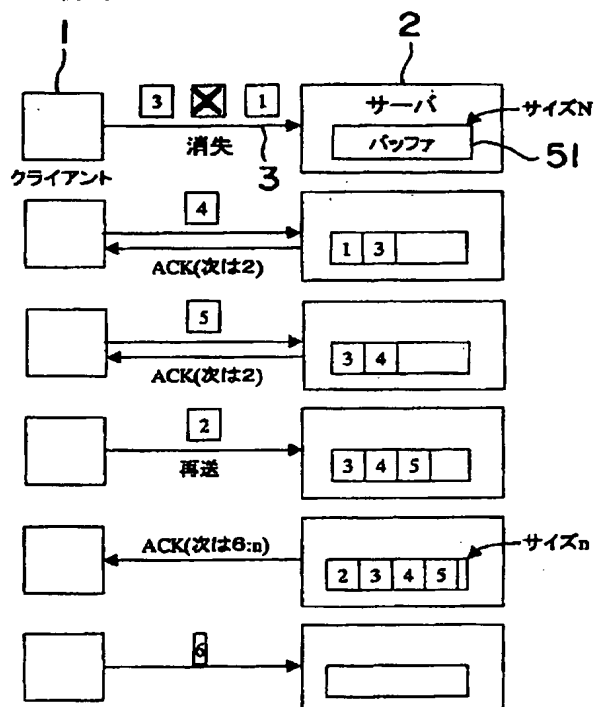
【図7】

負荷検出においてデータサイズによって対象データとするか否かの弁別を行う例を示す説明図



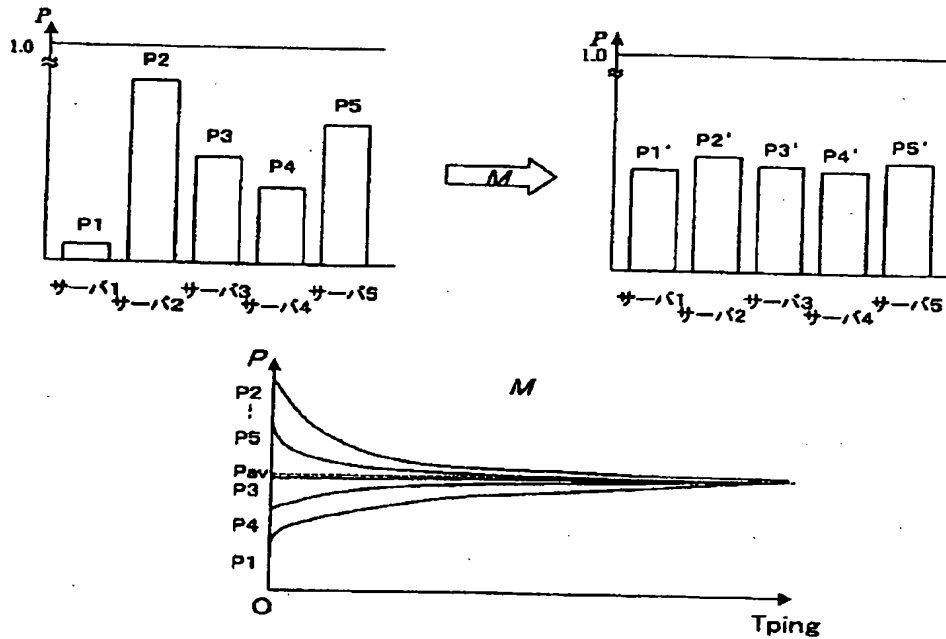
【図8】

シーケンス番号に基づくサーバの処理を説明するための図



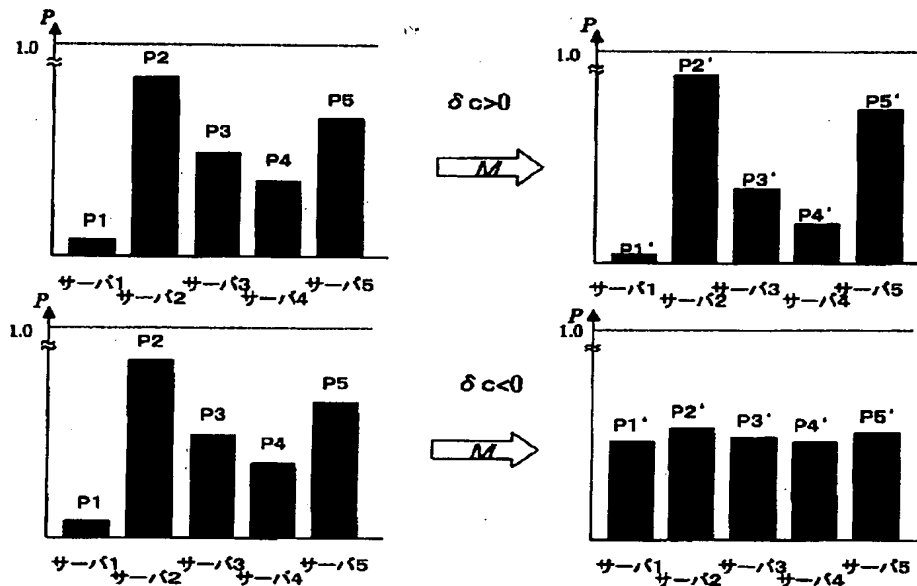
【図12】

修正関数を説明するための図(1)



【図15】

修正関数を説明するための図(2)



フロントページの続き

(72)発明者 横山 乾
神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(72)発明者 菊池 慎司
神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

Fターム(参考) 5B045 BB28 BB42 GG05
5B089 GB02 JA11 KA06 MC06
5K030 LB02 LB05 LC01 LC11
5K033 CB01 CB06